

VŠB – Technická univerzita Ostrava  
Fakulta elektrotechniky a informatiky  
Katedra informatiky

**Analýza držení volantu s využitím  
kamery umístěné ve vozidle**

**Analysis of Steering Wheel Holding  
using In–Car Camera**

## Zadání bakalářské práce

Student: **Tomáš Anlauf**

Studijní program: B2647 Informační a komunikační technologie

Studijní obor: 2612R025 Informatika a výpočetní technika

Téma: **Analýza držení volantu s využitím kamery umístěné ve vozidle**  
**Analysis of Steering Wheel Holding using In-Car Camera**

Jazyk vypracování: čeština

### Zásady pro vypracování:

Správné držení volantu je předpokladem pro bezpečnou jízdu a schopnost rychlého manévrování v případě náhlé nebezpečné situace. Cílem této bakalářské práce je detekovat ruce řidiče a analyzovat jejich pozici vůči volantu. K detekci poslouží kamera umístěná ve vozidle nad volantem. Tento systém pak může řidiče informovat o nesprávném způsobu držení.

Ve své práci proveďte:

1. Popište zadaný problém.
2. Analyzujte řešení a popište potřebnou teorii.
3. Proveďte vlastní implementaci řešení a její testování v reálných podmínkách.
4. Zhodnoťte dosažené výsledky.

### Seznam doporučené odborné literatury:


- [1] Zhou et al.: Hierarchical Context-Aware Hand Detection Algorithm for Naturalistic Driving. 2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC), pp. 1291-1297
- [2] Ohn-Bar et al.: Head, Eye, and Hand Patterns for Driver Activity Recognition. 2014 22nd International Conference on Pattern Recognition, pp. 660-665.
- [3] H. N. Le et al.: Robust Hand Detection and Classification in Vehicles and in the Wild. 2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), pp. 1203-1210.

Formální náležitosti a rozsah bakalářské práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí bakalářské práce: **Ing. Michael Holuša**

Datum zadání: 01.09.2017

Datum odevzdání: 30.04.2018

  
doc. Ing. Jan Platoš, Ph.D.  
vedoucí katedry



  
prof. Ing. Pavel Brandštetter, CSc.  
děkan fakulty

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně. Uvedl jsem všechny literární  
prameny a publikace, ze kterých jsem čerpal.

V Ostravě 30. dubna 2018

.....  
*Anlauf*

Rád bych poděkoval svým rodičům, p. Ing. Michaeli Holušovi a p. Ing. Radovanovi Fuskovi Ph.D., kteří mi poskytli podporu a měli se mnou trpělivost při tvorbě této práce.

## **Abstrakt**

V této práci je popsána metoda pro detekci a určení polohy rukou na volantů na základě zpracování obrazu z videozáznamu. Jde především o určení, zda ruce jsou ve správné bezpečné poloze na volantů. Implementace této metody je založena na segmentaci oblastí s barvou kůže z obrazu, s využitím open-source knihovny pro zpracování obrazu OpenCV. Jednotlivé sekce práce slouží k seznámení s barevnými modely, použitými technikami, popisem implementace a otestování funkčnosti a zhodnocení její úspěšnosti.

**Klíčová slova:** OpenCV, Volant, Kůže, Detekce rukou

## **Abstract**

In this work, a method for hand detection and position determination on a steering wheel based on the image processing of the video is described. It primarily focuses on determining whether hands are in correct and safe position. Implementation of the method is based on skin region segmentation from image with use of opensource library for image processing called OpenCV. Individual sections serve to familiarize with colour models, used techniques, implementation description and to test the method's functionality and evaluate its success rate.

**Key Words:** OpenCV, Steering Wheel, Skin, Hand detection

# Obsah

<b>Seznam použitých zkratk a symbolů</b>	<b>7</b>
<b>Seznam obrázků</b>	<b>8</b>
<b>Seznam tabulek</b>	<b>9</b>
<b>1 Úvod</b>	<b>10</b>
<b>2 Barevné modely</b>	<b>11</b>
2.1 RGB Model . . . . .	11
2.2 HSV Model . . . . .	12
2.3 YCbCr Model . . . . .	13
<b>3 Použité metody</b>	<b>14</b>
3.1 Hlubkový obraz . . . . .	14
3.2 Prahování . . . . .	14
3.3 Vzdálenostní transformace . . . . .	16
3.4 Watershed transformace . . . . .	16
3.5 Kontury . . . . .	17
<b>4 Implementace systému</b>	<b>19</b>
4.1 OpenCV . . . . .	19
4.2 Vstupní data . . . . .	19
4.3 Přehled systému . . . . .	21
4.4 Odstranění pozadí . . . . .	21
4.5 Detekce barvy kůže . . . . .	22
4.6 Segmentace obrazu pomocí Watershed algoritmu . . . . .	25
4.7 Vyhledání rukou . . . . .	27
4.8 Určení pozice rukou . . . . .	28
<b>5 Experimenty a výsledky</b>	<b>31</b>
<b>6 Závěr</b>	<b>35</b>
<b>Literatura</b>	<b>36</b>
<b>Přílohy</b>	<b>37</b>

## Seznam použitých zkratek a symbolů

RGB	– Red, Green, Blue
CMY	– Cyan, Magenta, Yellow
HSV	– Hue, Saturation, Value
YCrCb	– Luminance, Red chromation component, Blue chromation component
OpenCV	– Open-source Computer Vision
LCD	– Liquid-crystal display
JPEG	– Joint Photographic Experts Group
MPEG	– Moving Picture Experts Group
PGM	– Portable Gray Map

## Seznam obrázků

1	RGB barevný model [1]	12
2	HSV barevný model [2]	13
3	YCrCb barevný model [3]	13
4	Vstupní obrázky	20
5	Struktura programu pro detekci rukou	21
6	Výsledný obraz po odstranění pozadí	22
7	Výsledek segmentace kůže	24
8	Aplikace morfologických operací	25
9	Jednotlivé kroky při aplikaci Watershed transformace	28
10	Ukázka správné detekce rukou při jejich správné pozici	32
11	Ukázka správné detekce rukou při jejich špatné pozici	33
12	Ukázka chybných detekcí nebo špatné určení polohy rukou	34



## Seznam tabulek

1	Nastavené statické hodnoty prahu a hloubky . . . . .	31
2	Přesnost detekce . . . . .	31

# 1 Úvod

Detekce rukou je důležitým faktorem pro porozumění aktivity řidiče během jízdy, jelikož ruce jsou běžná média pro vyjádření informací o řidiči. Úspěšné rozpoznání lidské aktivity má velký potenciál pro aplikace ve vozidle. Například bude možné upozornit na aktivity, které mohou způsobit ztrátu pozornosti řidiče. Pokud ruce nejsou na volantu ve správné poloze nebo nejsou na volantu vůbec, pak může u řidičů dojít k nedostatečně rychlé reakci na náhlé nebezpečné situace a pravděpodobnost nehody se tedy zvyšuje. Zároveň by mohlo být možné využít rozpoznání rukou v oblasti volantu v samořízených autech pro aktivaci režimu samořízení, pokud v dané oblasti volantu nebudou ruce nalezeny.

Existují hlavní dva přístupy pro detekci rukou na volantu: metody založené na doteku a detekce v obrazu [4]. Metody založené na doteku detekují ruce na základě fyzické interakce mezi rukama a hlavními komponenty pro řízení automobilu (volant, řadící páka, ruční brzda atd.). I přesto že jsou považovány za spolehlivější metody detekce, tak nemohou detekovat ruce, pokud nejsou v přímém doteku s komponenty vozidla. Detekce v obrazu spoléhá na kamery, které zachycují interiér vozidla a algoritmy zpracování obrazu k detekci rukou. Tyto metody jsou schopné detekovat ruce v celém zorném poli kamery, ovšem s nižší přesností.

Detekce rukou v přirozeném prostředí sebou přináší mnohé výzvy, jako jsou změny v celkovém osvětlení, pohyb stínů, změny v tvaru ruky, barvy objektů, které jsou podobné barvě lidské kůže. Existuje mnoho různých metod detekce rukou. Pro vyřešení zadaného úkolu byla vybrána detekce v obraze založená na segmentaci oblastí s barvou kůže. V této práci je tedy popsán a navržen systém, který má za úkol vyřešit tyto problémy a dosáhnout tak maximálního správného výsledku detekce rukou.

První kapitola se zaměřuje na barevné systémy, které jsou použity při tvorbě pravidel pro segmentaci kůže z obrazu. Druhá kapitola popisuje segmentační metodu prahování a hloubkové obrazy, které jsou použity pro odstranění pozadí. Třetí kapitola obsahuje informace o další segmentační metodě s názvem Watershed, která je použita pro oddělení objektů v obraze. Čtvrtá kapitola se zaměřuje na kontury, které jsou použity pro označení hranic rukou a pomocí jejich vlastností je určena jejich poloha. Pátá kapitola popisuje samotnou implementaci systému pro detekci rukou a určení polohy. Šestá kapitola obsahuje výsledky detekce pomocí navrženého systému.

## 2 Barevné modely

Barevný model je matematický model, který reprezentuje informaci o barvě jako tři nebo čtyři různé barevné složky. Snaží se napodobit co nejlépe barvy v přírodě, které vznikají směsí různých vlnových délek světla. Barevné modely popisují základní barvy a poté i model, sloužící pro míchání těchto základních barev do výsledné barvy. Nejčastější použití mají ty barevné modely, u kterých není model příliš složitý a mají věrné podání barvy. Mezi základní zástupce barevných modelů patří barevné modely: **RGB** (Red – červená, Green – modrá, Blue – modrá), **CMYK** (Cyan – azurová, Magenta – purpurová, Yellow – žlutá, black – černá), používaná pro tisk jako zástupce subtraktivního míchání barev, **HSV** (Hue – barva, Saturation – sytost, Value – hodnota jasu) a model YCbCr (Y – luminance, Cb a Cr – chrominancií komponenty), který je odvozen od RGB modelu. [5] Barevné modely mají mnoho oblastí použití jako jsou počítačová grafika, zpracování obrazu, televizní vysílání atd. [3]

### 2.1 RGB Model

Jedním ze základních barevných modelů je barevný model RGB. Skládá se ze tří základních barev, kterými jsou červená(R), zelená(G) a modrá(B). Je zástupcem aditivního míchání barev, kdy jednotlivé složky se sčítají a vytvářejí světlo vyšší intenzity. Kombinace všech složek při maximální hodnotě vytvoří barvu bílou[6]. Barvy mohou být reprezentovány vektorem, kde hodnota každé složky se pohybuje v rozmezí  $\langle 0, 1 \rangle$ . Obvykle se každá složka kóduje do jednoho bytu a nabývá tak hodnot od 0 do 255. Počet odstínů, které lze tedy vytvořit, se rovná  $256^3 = 16777216$ . Obrázek 1 zobrazuje, že barevný model RGB lze reprezentovat krychlí, která má v počátku souřadnic černou barvu  $[0,0,0]$  a v protilehlém vrcholu barvu bílou  $[1,1,1]$  [5]. Tento model je široce využíván pro uchování a reprezentaci digitálního obrazu. Je používán v počítačích, grafických kartách a LCD monitorech[3].

Získání podílu jednotlivých složek je zobrazeno v Rovnici (1) až Rovnici (3)[3].

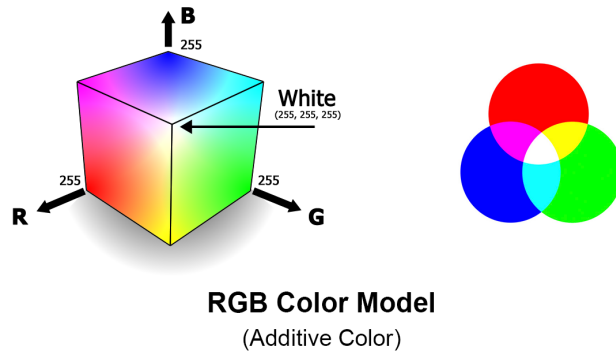
$$r = \frac{R}{R + G + B} \quad (1)$$

$$g = \frac{G}{R + G + B} \quad (2)$$

$$b = \frac{B}{R + G + B} \quad (3)$$

Při převodu barevného modelu RGB na šedotónový se využívá vztahu pro výpočet jasu, založeného na principu, že lidské oko vnímá různě stejnou intenzitu různých barev. Vztah pro výpočet intenzity je uveden v Rovnici(4) [5].

$$I = 0,299R + 0,587G + 0,114B \quad (4)$$



Obrázek 1: RGB barevný model [1]

## 2.2 HSV Model

HSV model je více intuitivní pro člověka než RGB model a proto je často používán pro výběr barev v grafických aplikacích. Model je možné prezentovat jako kužel, který je vhodný, protože zobrazuje všechny barvy v jednom objektu [5]. Pro vytvoření barvy slouží tři rozměry. Barva(H) nabývá hodnoty od 0 do 360. Barva je rozdělena na šest částí, kde v každé můžeme najít odstíny, vyskytující se mezi dvěma krajními barvami. Tyto krajní barvy jsou červená, žlutá, zelená, modrozelená, modrá, purpurová. Hodnoty barvy znázorňují úhel na kruhové ploše a proto se při překročení hodnoty 360 barvy opakují. Sytost(S) nabývá hodnoty od 0,0 do 1,0, čímž se mění úroveň šedi dané barvy. Hodnota v kuželu značí vzdálenost od středu kruhové podstavy k jejímu okraji. Jas(V) nabývá hodnot od 0 do 1,0 a znázorňuje vzdálenost od středu kruhové podstavy až ke špičatému konci kuželu [7] [3]. Na Obrázku 2 je graficky znázorněn HSV model.

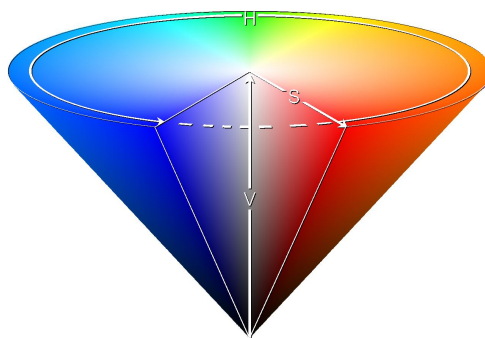
Pro převod z RGB na HSV slouží algoritmus[5], v němž jednotlivé komponenty RGB modelu nabývají hodnot  $r, g, b \in \langle 0, 1 \rangle$  a jedna složka bude v maximu a zároveň jedna v minimu, tak pro výpočet barevného úhlu  $h \in \langle 0, 360 \rangle$  slouží Rovnice (5):

$$h = \begin{cases} 0 & \text{if } \max = \min \\ \left(60^\circ \cdot \frac{g-b}{\max-\min} + 360^\circ\right) \bmod 360^\circ & \text{else if } \max = r \\ \left(60^\circ \cdot \frac{b-r}{\max-\min} + 120^\circ\right) & \text{else if } \max = g \\ \left(60^\circ \cdot \frac{r-g}{\max-\min} + 240^\circ\right) & \text{else if } \max = b \end{cases} \quad (5)$$

Pro výpočet sytosti a hodnoty jasu  $s, v \in \langle 0, 1 \rangle$  použijeme následující dvě Rovnice (6) a (7):

$$s = \begin{cases} 0 & \text{if } \max = 0 \\ \frac{\max-\min}{\max} & \text{else} \end{cases} \quad (6)$$

$$v = \max \quad (7)$$

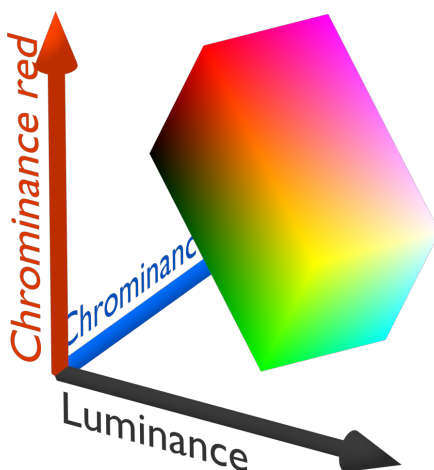


Obrázek 2: HSV barevný model [2]

### 2.3 YCbCr Model

YCbCr model je zakódovaný nelineární model, odvozený od RGB modelu. Je typicky využíván v analogových a digitální videích. Jelikož tento model umožňuje lehce se zbavit redundantních informací o barvě, je často používán ve standardech pro kompresi obrázků a videí jako jsou JPEG, MPEG1, MPEG2, MPEG4. Skládá se z luminance (Y) a dvou chrominancí (Cb a Cr), což je ukázáno na Obrázku 3. Cb reprezentuje rozdíl mezi modrým komponentem a referenční hodnotou luminance. Cr reprezentuje rozdíl mezi červeným komponentem a referenční hodnotou luminance [3]. Vypočet jednotlivých komponentů je ukázán v Rovnici (8) [8].

$$\begin{bmatrix} Y \\ Cr \\ Cb \end{bmatrix} = \begin{bmatrix} 0 \\ 128 \\ 128 \end{bmatrix} + \begin{bmatrix} 0,299 & 0,587 & 0,114 \\ -0,169 & -0,331 & 0,500 \\ 0,500 & -0,419 & -0,081 \end{bmatrix} \cdot \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad (8)$$



Obrázek 3: YCrCb barevný model [3]

## 3 Použité metody

Mimo segmentaci barvy kůže jsou v programu použity další segmentační metody, které slouží pro zpracování obrazu. První z těchto metod je prahování. Prahování je využito pro odstranění pozadí z obrazu a předzpracování obrazu. Další segmentační metodou je Watershed transformace, která je použita pro oddělení navzájem blízkých oblastí rukou. Poslední metodou je vyhledání kontur, sloužící pro získání oblastí rukou, které v obraze zůstaly po jeho zpracování. Tyto metody jsou dále popsány podrobněji.

### 3.1 Hloubkový obraz

Hloubkový obraz je typ obrázku, který obsahuje informace o vzdálenosti povrchu objektů od kamery. Vzdálenost je určena pomocí intenzity barvy pixelů. Se zvyšující se vzdáleností od místa pohledu se intenzita barvy zvyšuje. Existují ovšem i zařízení, které poskytují hloubkový obraz s opačnou reprezentací vzdálenosti, tzn. se zvyšující se vzdáleností se intenzita barvy snižuje. Místa v obraze, ve kterých vzdálenost není definována, jsou obvykle reprezentovány černou barvou. Intenzity barev nabývají hodnoty od 0 do maximální hodnoty podle velikosti obrázku. Pro 8-bitový obraz je maximální hodnota 255 a pro 16-bitový obraz je to 65 535. [9].

### 3.2 Prahování

Prahování neboli threshold je nejjednodušší segmentační metoda, která se používá k oddělení objektů od pozadí. V nekomplikovaných případech je tato metoda současně i rychlá a spolehlivá. Každému pixelu je přidělena intenzita barvy. Výsledkem je binární obraz, v němž je jednotlivým bodům nalezených oblastí přiřazena hodnota 1, zatímco bodům pozadí je přiřazena hodnota 0. Při prahování se vychází z předpokladu, že body vyhledávaných oblastí mají stejný nebo velmi podobný jas. Při prahování je pixel označen jako bod oblasti, jestliže svým jasnem spadá do navoleného intervalu  $\langle a, b \rangle$ . Velmi často je při rozhodování použita jediná hodnota  $t$  – tzv. práh. Body s jasnem vyšším než je hodnota prahu jsou označeny jako body oblasti a body, jejichž jas je nižší než hodnota prahu, jsou označeny jako body pozadí (popřípadě naopak). Úspěšnost tedy závisí na znalosti správné hodnoty prahu[10].

Existuje pět základních typů prahování: Binární (Binary), Invertované binární (Binary, Inverted), Seřezávací (Truncate), Do nuly (To Zero), Invertované do nuly (To Zero, Inverted). Tři z těchto uvedených typů byly využity v programu a budou následně popsány.

#### 3.2.1 Binární prahování

Pokud je intenzita pixelu vstupního obrazu na pozici  $x, y$  ( $\text{src}(x, y)$ ) větší než hodnota prahu, pak je intenzita nového pixelu výstupního obrazu na pozici  $x, y$  ( $\text{dst}(x, y)$ ) nastavena na maximální

intenzitu. Pokud nastane opak, tak je intenzita nastavena na nulu. Tato operace je vyjádřena v Rovnici(9).

$$dst(x, y) = \begin{cases} maxHod & \text{if } src(x, y) > prahovacíHodnota \\ 0 & \text{else if } src(x, y) \leq prahovacíHodnota \end{cases} \quad (9)$$

### 3.2.2 Prahování do nuly

Pokud je intenzita pixelu vstupního obrazu na pozici x,y ( $src(x, y)$ ) nižší než hodnota prahu, pak je intenzita nového pixelu výstupního obrazu na pozici x,y ( $dst(x, y)$ ) nastavena na nulu. Tato operace je vyjádřena v Rovnici(10).

$$dst(x, y) = \begin{cases} src(x, y) & \text{if } src(x, y) > prahovacíHodnota \\ 0 & \text{else if } src(x, y) \leq prahovacíHodnota \end{cases} \quad (10)$$

### 3.2.3 Invertované prahování do nuly

Pokud je intenzita pixelu vstupního obrazu na pozici x,y ( $src(x, y)$ ) větší než hodnota prahu, pak je intenzita nového pixelu výstupního obrazu na pozici x,y ( $dst(x, y)$ ) nastavena na nulu. Tato operace je vyjádřena v Rovnici(11).

$$dst(x, y) = \begin{cases} 0 & \text{if } src(x, y) > prahovacíHodnota \\ src(x, y) & \text{else if } src(x, y) \leq prahovacíHodnota \end{cases} \quad (11)$$

### 3.2.4 Otsuovo prahování

Tato metoda rozděluje obraz na popředí a pozadí automatickým nalezením hodnoty prahu. Toto automatické nalezení hodnoty prahu může být použito pro obraz s dvěma či více vrcholy ve svém histogramu. Přes tyto hodnoty se iteruje a zjišťuje se maximální mezitřídní rozptyl, podle kterého se poté určí hodnota prahu. S přibývajícím počtem vrcholů klesá přesnost stanovení hodnoty pro rozdělení popředí a pozadí, proto je nejvhodnější tuto metodu použít pro obraz s histogramem, který obsahuje pouze dva vrcholy nebo-li bimodální obraz [11].

### 3.2.5 Statické určení hodnoty prahu

Hodnota prahu, která se pro 16-bitový obrázek pohybuje v hodnotě od 0 do 65 535, je vybrána na základě specifikací uživatele. Pro každé použití prahování v systému je proto potřeba, aby uživatel nastavil optimální hodnotu pro prahování tak, aby byl zajištěn žádoucí výsledek. S různými vozidly a jejich vzdáleností od volantu k místům nad čelním sklem se může hodnota lišit. Tudíž tato metoda nemusí být vždy efektivní, jelikož úspěch či selhání systému je závislé na přesnosti uživatelsky zvolené hodnoty[12].

Jednotlivé navolené hodnoty, které byly použity při prahování pro každé video, jsou popsány v tabulce 1 v Sekci 5. Hodnoty byly navoleny tak, aby se odstranila co největší část pozadí a přitom se v obraze zachovala spodní část volantu.

### 3.3 Vzdálenostní transformace

Vzdálenostní transformace je metoda, která je nejčastěji aplikována na binární obrazy. Výsledkem transformace je šedotónový obraz, který se liší od vstupního tím, že intenzita barvy každého pixelu objektu v popředí je změněna podle vzdálenosti k nejbližšímu pixelu pozadí. Existují tři způsoby, jak takovouto vzdálenost  $D$  vypočítat pro dva pixely se souřadnicemi  $(x_1, x_2)$  a  $(y_1, y_2)$  [13].

Prvním je výpočet vzdálenosti pomocí Euklidovské vzdálenosti uvedený v Rovnici (12).

$$D = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \quad (12)$$

Druhým způsobem je Manhattanská vzdálenost, která předpokládá, že můžeme jít z jednoho pixelu do druhého pouze ve směru os pixelové mřížky. Výpočet je uveden v Rovnici (13).

$$D = |x_2 - x_1| + |y_2 - y_1| \quad (13)$$

Posledním způsobem je určení Šachovnicové vzdálenosti, při které se na pixelové mřížce můžeme pohybovat jako král při hraní šachů. Výpočet je uveden v Rovnici (14).

$$D = \max(|x_2 - x_1|, |y_2 - y_1|) \quad (14)$$

### 3.4 Watershed transformace

Watershed transformace je technika, používaná pro oddělení objektů v obraze od pozadí, jako například rozdělení obrazu na samostatné oblasti, kde každá oblast je homogenní s ohledem na vlastnost, jakou je například intenzita barvy nebo textura [14]. Technika používá metodu pro růst jednotlivých oblastí v obraze. Princip je převzat z geografie, kde jednotlivé oblasti představují vodní nádrž. V této topografické interpretaci mohou existovat tři typy bodů[15]:

1. Body patřící lokálním minimům oblastí.
2. Body, kde umístěná kapka vody s jistotou steče do jednoho lokálního minima.
3. Body, kde voda by mohla se stejnou pravděpodobností stéct do více lokálních minim.



Body, které spadají do první kategorie, mohou být nazývány dnem nádrže. Body, splňující podmínku druhé kategorie, mohou být nazývány oblastí nádrže. Nakonec body, splňující podmínku třetí kategorie, mohou být nazývány jako hráz vodní nádrže. Pro simulaci tohoto procesu při segmentaci obrazu mohou být použity dva přístupy: buďto najít první oblast nádrží a následně hráze pomocí vytvoření doplňku oblasti nádrže nebo vypočítat celkové rozdělení obrazu do nádrží a poté najít hráze pomocí detekce hran[14].

Hlavními výhodami Watershed metody jsou [15]:

1. Výsledné hráze tvoří uzavřené a spojitě oblasti. Tradiční metody detekce hran často vytvoří nespojitě hrany, které je potřeba dále zpracovat k jejich uzavření.
2. Hráze výsledných oblastí se vždy pojí s konturami, které se v obraze objevují jako očividné kontury objektů.
3. Sjedení všech výsledných oblastí vytvoří celou oblast obrazu.

Hlavní nevýhodou je, že pro většinu přirozených obrázků metoda produkuje nadměrnou segmentaci, která způsobí vznik velkého množství malých segmentovaných oblastí v obraze.

### 3.5 Kontury

Kontury reprezentují hranice objektu v obraze. Lze je reprezentovat buďto uspořádaným seznamem hran nebo křivkou. Kontury mohou být otevřené nebo zavřené. Uzavřené kontury odpovídají ohraničení oblasti a pixely v této oblasti mohou být určeny pomocí vyplňujícího algoritmu. Otevřené kontury mohou být součástí uzavřené oblasti. Mezery v hraniční oblasti se mohou objevit, pokud kontrast mezi oblastmi není dostatečný pro to, aby vznikl ostrý přechod, který může být detekován. Nalezené kontury lze dále využít pro zjištění počtu oddělených objektů v obraze, kategorizaci tvarů objektů, vykrojení objektů z obrazu atd.

Pokud jsou vnitřní detaily objektu viditelné v obrázku, tak objekt může obsahovat několik souvisejících vnitřních kontur, které jsou uspořádány do hierarchie. Hierarchie pro každou nejsvrchnější konturu je znázorněna pomocí stromu s nejsvrchnější konturou jako kořenem stromu. Při pohybu hierarchií směrem k listům, každá úroveň ve stromu reprezentuje další stále vnitřnější kontury. Hierarchie kontury je v OpenCV reprezentována jako pole všech kontur, kde pro každou konturu je vytvořeno pole čtyř hodnot, které reprezentují indexy sousedních kontur. První hodnota ukazuje na další konturu na stejné úrovni v hierarchii, druhá ukazuje na předchozí, třetí na prvního potomka a čtvrtá na předka.

Další informace o objektu můžeme získat pomocí jeho příznaků. Příznaky jsou číselné hodnoty, které popisují každý objekt v obraze. Jedním z těchto příznaků jsou momenty různého stupně, které jsou velmi často využívány. Jsou využívány proto, jelikož jejich schopnost od sebe odlišit objekty různých tříd často vyhovuje a jejich výpočet je snadný. Výpočet momentu vztaženého k souřadné soustavě obrazu je definován v Rovnici (15)[10].

$$m_{p,q} = \iint_{\Omega} x^p y^q f(x, y) dx dy \quad (15)$$

kde  $f(x, y)$  je obrazová funkce,  $p, q$  jsou stupně momentu,  $\Omega$  je část obrazu, považovaná za rozpoznaný objekt. Hodnoty  $p, q$  lze volit  $p \geq 0, q \geq 0$ . Moment  $m_{0,0}$  je plocha objektu vážená hodnotou jasu. Zbývající momenty  $m_{p,q}$  jsou závislé na poloze rozpoznávaného objektu v obraze, většinou jsou tyto momenty vypočteny vzhledem k osám procházejícím těžištěm objektu.

## 4 Implementace systému

Cílem je navrhnout, popsat a implementovat program schopný efektivního rozeznání polohy rukou řidiče vzhledem k pozici volantu. Takto získané informace mohou být využity pro zjištění stavu či chování řidiče automobilu, případně mohou sloužit k upozornění řidiče na jeho špatné držení volantu a tím zvýšit jeho bezpečnost při jízdě. Program byl vytvořen v programovacím jazyce C++.

### 4.1 OpenCV

OpenCV (Open Source Computer Vision Library) je knihovna, jejíž hlavní zaměření je zpracování obrazu. Ze zpracovaného obrazu jsme schopni získat relevantní informace. Tato knihovna byla vyvinuta společností Intel. Knihovna je nyní volně šiřitelná pro vývoj a komerční použití. Je napsána v programovacím jazyce C/C++, je nezávislá na platformě a proto je možné s ní pracovat na mnoha operačních systémech jako je Windows, Linux atd. Nabízí plně funkční rozhraní nejen pro jazyk C/C++, ale i pro jazyky Python, Java, MATLAB a Octave. Knihovna nám nabízí základní datové struktury pro operace s maticemi a pro zpracování obrazu. Také poskytuje rychlé, efektivní a spolehlivé algoritmy, které mohou být použity v komplexních aplikacích pro zpracování obrazu v reálném čase[5] [12].

### 4.2 Vstupní data

Pro testování systému byly vytvořeny datasety obrázků, které jsou uspořádány dle času a data pořízení a v posloupnosti za sebou tvoří video. Datasets jsou kolekce dat, v tomto případě obrázků, které sdílí společné vlastnosti a lze k nim přistupovat jednotlivě nebo s nimi pracovat jako s jedním celkem.

Získané datasety obsahují dva typy obrázků, jimiž jsou RGB barevné obrázky ve formátu JPEG a hloubkové obrázky ve formátu PGM. Navržený systém využívá k detekci oba typy obrazu. Pro zajištění přesnosti a zároveň funkčnosti systému jsou hloubkové obrázky používány ve dvou formách, jednou jako 8-bitový obraz a podruhé jako 16-bitový, jelikož řada funkcí knihovny OpenCV není schopna pracovat s 16-bitovým obrazem.

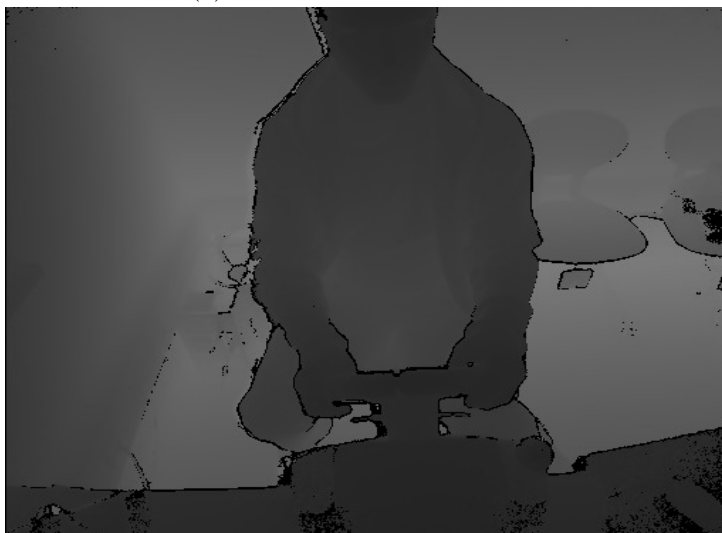
Před hlavním zpracováním obrazu pro detekci rukou je potřeba vstupní obrázku předzpracovat. Barevný obraz je zpracován vyrovnáním histogramu. Vyrovnání histogramu je bodová operace, která zajistí rovnoměrné rozložení hustoty pravděpodobnosti jasu výsledného obrazu. Pomocí této operace je obraz transformován a jsou zajištěny normalizované jednotné podmínky před dalším zpracováním [10]. V OpenCV je tato funkce k dispozici jako *equalizeHist*. Jelikož funkce může zpracovat pouze jednu barevnou složku, je potřeba rozdělit barevný obrázek na jednotlivé barevné složky, na každou zvlášť tuto funkci aplikovat a poté barevné složky spojit zpět do RGB obrázku.

Dále je vyznačena v obraze oblast volantu, která se bude zpracovávat. Tento krok slouží k tomu, aby byla zpracována pouze nezbytná část obrazu a zároveň k eliminaci oblastí v interiéru vozidla, které mají barvu kůže a mohly by tak negativně ovlivnit výsledek detekce. Je důležité, aby se volant nacházel uprostřed této vyznačené oblasti pro pozdější určení pozice rukou vzhledem k tomuto volantu.

Na Obrázku 4 jsou ukázány příklady vstupních obrázků, na kterých budou demonstrovány jednotlivé aplikace částí systému.



(a) Příklad barevného obrázku

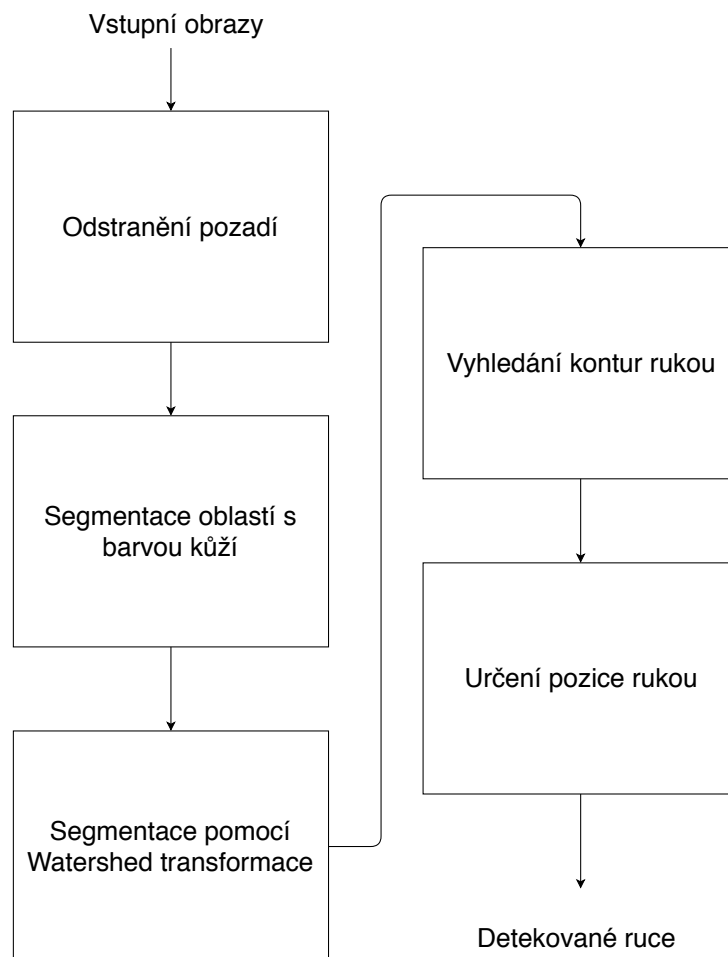


(b) Příklad hloubkového obrázku

Obrázek 4: Vstupní obrázky

### 4.3 Přehled systému

Detekce rukou na volantě je rozdělena do několika částí. Struktura programu pro detekci rukou je znázorněna v diagramu na Obrázku 5.



Obrázek 5: Struktura programu pro detekci rukou

### 4.4 Odstranění pozadí

V této části implementace je použita technika, pomocí které je odstraněna většina objektů v pozadí, které by mohly být považovány za ruce a tak vést ke špatnému výsledku. Touto technikou je **prahování**. Jako vstupní data poslouží 16-bitový hloubkový obraz interiéru vozidla, na který bude tato technika aplikována a podle výsledku budou ovlivněny i ostatní obrázky se vstupními daty.

Pro odstranění pozadí je využito metody invertovaného prahování do nuly. Hodnota prahu je navolena staticky tak, aby zachovala oblast volantu a zároveň odstranila, co největší část pozadí. Stanovené hodnoty prahu pro jednotlivé vstupní videa jsou popsány v Sekci 5.

Na Obrázku 6 můžeme vidět výsledek prahování, po kterém ve výstupním obrazu zůstávají objekty pouze v popředí ve vzdálenosti volantu.



Obrázek 6: Výsledný obraz po odstranění pozadí

#### 4.5 Detekce barvy kůže

Výběr barevného modelu je základním procesem pro navržení modelu pro segmentaci barvy kůže. Jeden nebo více modelů může poskytnout optimální hodnoty pro detekce pixelů s kůží ve vstupním obrazu. Výběr barevného modelu často závisí na metodě, která je použita pro detekci kůže.

V této části programu je řešeno získání oblastí s barvou kůže ze vstupních RGB obrázků. Tyto oblasti jsou získány pomocí modelu sestaveného ze tří barevných modelů RGB, HSV a YCrCb. Pro každý z těchto barevných systémů jsou sestavena pravidla, která určují hranice založené na rozsahu jednotlivých složek definujících barvu kůže. V prvním kroku jsou vstupní RGB obrázky převedeny na barevné systémy HSV a YCrCb. V druhém kroku jsou tato pravidla použita k segmentaci kůže ze vstupního obrazu. V posledním kroku je použita kombinace morfologických operací, které zajišťují odstranění oblastí, které nejsou součástí kůže řidiče a naopak také pro zlepšení oblastí, které náleží rukám řidiče.

Před samotnou aplikací pravidel pro získání oblastí s kůží je potřeba vstupní RGB obraz převést na ostatní barevné modely. Byly vytvořeny obrazy, ve kterých jsou barvy pixelů reprezentovány modely HSV a YCrCb. Obraz s barevným modelem HSV je dále nutné pro správnou funkčnost převést na 32-bitový obraz s přesností datového typu float, pro zajištění práce s celým spektrem složky Hue (H) od 0 až po 360. Dále je HSV obrázek normalizován a tím je změněn rozsah hodnot na 0 až 255. Pro všechny převody mezi barevnými modely, změnu formátu obrazu a normalizaci byly využity metody, které nabízí knihovna OpenCV.

#### 4.5.1 Omezující pravidla pro barvu kůže

Při analýze barevných modelů pro získání modelu barvy kůže byly odvozeny omezující pravidla pro barevné modely RGB, HSV a YCbCr [16].

V RGB modelu byly použity pro segmentaci kůže pravidla představena v [17]. Pravidla jsou odvozena pro hodnoty intenzit v rozmezí 0 až 255. Barva kůže, osvětlena stejnoměrným slunečním světlem, je získána pomocí pravidel popsanych v Rovnici (16):

$$\begin{aligned} & (R > 95) \wedge (G > 40) \wedge (B > 20) \quad \wedge \\ & (maxR, G, B - minR, G, B > 15) \quad \wedge \\ & (|R - G| > 15) \wedge (R > G) \wedge (R > B) \end{aligned} \quad (16)$$

Barva kůže, osvětlena svítlnou či postranním denním světlem, je definována v Rovnici (17):

$$\begin{aligned} & (R > 220) \wedge (G > 210) \wedge (B > 170) \quad \wedge \\ & (|R - G| \leq 15) \wedge (R > B) \wedge (G > B) \end{aligned} \quad (17)$$

Pro segmentaci barvy kůže nám postačí, pokud bude splněno alespoň jedno z výše zmíněných pravidel, a proto použijeme logický součet pro zkombinování obou pravidel, což je znázorněno v Rovnici (18).

$$Rovnice(16) \vee Rovnice(17) \quad (18)$$

Na základě průzkumu [16] bylo zjištěno, že podmodel CbCr model je silný diskriminant barvy kůže a bylo odvozeno těchto pět omezujících pravidel popsanych v Rovnici (19) až (23):

$$Cr \leq 1,5862 \cdot Cb + 20 \quad (19)$$

$$Cr \geq 0,3448 \cdot Cb + 76,2069 \quad (20)$$

$$Cr \geq -4,5652 \cdot Cb + 234,5652 \quad (21)$$

$$Cr \leq -1,15 \cdot Cb + 301,75 \quad (22)$$

$$Cr \leq -2,2857 \cdot Cb + 432,85 \quad (23)$$

Pravidla jsou poté zkombinována logickým součinem k vytvoření finálního CbCr pravidla, znázorněného v Rovnici (24).

$$Rovnice(19) \wedge Rovnice(20) \wedge Rovnice(21) \wedge Rovnice(22) \wedge Rovnice(23) \quad (24)$$

V modelu HSV, barevný tón(H) vykazoval nejzřetelnější hranici mezi oblastmi s barvou kůže a ostatními oblastmi. Dle průzkumu byla určena dvě pravidla [16], znázorněná v Rovnicích (25) a (26).

$$H < 25 \quad (25)$$

$$H > 230 \quad (26)$$

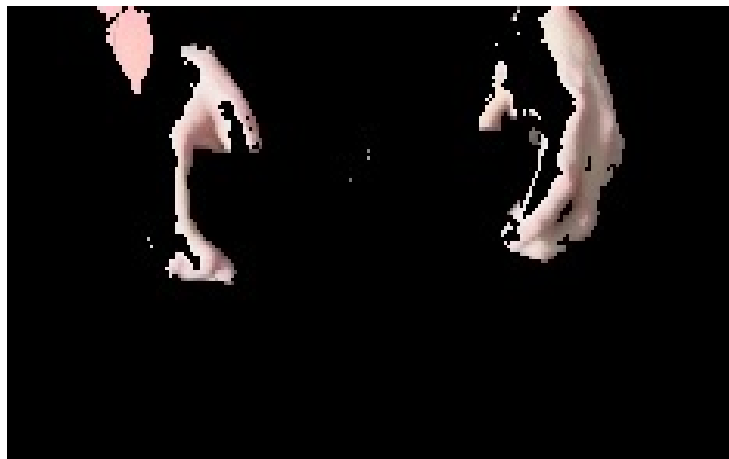
Obě pravidla jsou zkombinována logickým součtem k vytvoření finálního HSV pravidla, Rovnice (27)

$$Rovnice(25) \vee Rovnice(26) \quad (27)$$

Každý pixel, který splňuje finální rovnice každého barevného modelu (18), (24) a (27) je považován za pixel s barvou kůže.

#### 4.5.2 Výsledek segmentace kůže

Vytvořený model pro segmentaci oblastí s barvou kůže byl nazván RGB-H-CbCr model [16]. Tato segmentační technika je považována za nízkoúrovňovou extrakci a proto byla použita pouze pro zvýšení přesnosti při hledání rukou řidiče automobilu. Obrázek 7 ukazuje výsledek aplikované segmentace kůže.



Obrázek 7: Výsledek segmentace kůže



Ve výsledném obrazu se segmentovanou kůží mohou nastat tři problémy [16]:

1. Oblasti jsou fragmentovány a proto velmi často obsahují díry a mezery.
2. Pokud jsou oblasti rukou velmi blízko sebe, může poté dojít k chybné detekci rukou (např. ruce mohou být považovány za jednu ruku).
3. Extrahované oblasti s barvou kůže nemusejí býti nutně ruce řidiče. Oblasti mohou patřit obličejům a jiným odhaleným částem těla řidiče nebo jeho spolujezdců. Také interiér vozu a objekty v pozadí či popředí se můžou ve výsledku segmentace objevit, pokud se barevně blíží barvě kůže.

#### 4.5.3 Morfologické operace

Další krok zahrnuje použití morfologických operací k odstranění prvních dvou problémů popsaných v Sekci 4.5.2. K odstranění prvního problému je použita dilatace a eroze pro zacelení vzniklých mezer a děr. Druhý problém se nejčastěji projeví, pokud má řidič ruce příliš blízko sebe. Odstranění tohoto problému je zajištěno použitím morfologického otevření k rozdělení úzce spojených oblastí. Na Obrázku 8 je ukázána aplikace morfologických operací na obraz se segmentovanou kůží. Samotná segmentace kůže nestačí pro efektivní detekci rukou řidiče, jelikož vzniká vysoká pravděpodobnost, že ve výsledku budou jiné oblasti s barvou kůže kromě rukou řidiče. V následujících sekcích jsou uvedeny další kroky k dosažení správného výsledku.



Obrázek 8: Aplikace morfologických operací

#### 4.6 Segmentace obrazu pomocí Watershed algoritmu

Vstupním obrazem je barevný obraz se segmentovanými oblastmi kůže. Takto získané oblasti kůže musí být poté řádně odděleny a zvýrazněny v obraze pro snadnější a přesnější detekci rukou. K tomuto je využita Watershed transformace. Před samotnou aplikací transformace je nejdříve potřeba vstupní obraz několikrát předzpracovat, abychom dosáhli požadovaných výsledků.

Nejdříve jsou ve vstupním obraze zostřeny hrany nalezených oblastí. Tohoto je dosaženo pomocí aplikace Laplaceova filtru, který využije pro filtraci kernel s aproximací druhé derivace. Prvním krokem je tedy vytvoření tohoto kernelu, který je ukázán v Rovnici (28).

$$kernel = \begin{bmatrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{bmatrix} \quad (28)$$

Jelikož kernel obsahuje záporné hodnoty, je potřeba obraz převést na vhodný formát, kterým je 32-bitový formát s reprezentací intenzity pixelu pomocí datového typu `float`. Filtrace je provedena pomocí funkce implementované v knihovně OpenCV `filter2D`.

Požadovaný formát je pro tento případ nastaven jako `CV_32F`. Po skončení aplikace filtru je potřeba výsledný obraz zpět převést na formát 8-bitový s datovým typem `unsigned int` s třemi kanály pomocí funkce `convertTo` implementované v OpenCV, aby byla zajištěna funkčnost pro další zpracování.

Poté je upravený barevný obraz převeden na šedotónový pomocí funkce `cvtColor` implementované v OpenCV s příznakem `CV_BGR2GRAY`.

Takto převedený obraz je následně převeden na binární pomocí prahování. V tomto případě je použita Otsuova binarizace aplikovaná pomocí funkce `threshold`, která je součástí OpenCV. Pro zajištění Otsuovy binarizace je potřeba nastavit typ prahování jako `THRESH_BINARY + THRESH_OTSU`.

Následným zpracováním je aplikace vzdálenostní transformace. Tento operátor přijímá binární obraz jako vstupní a v tomto obraze změní intenzitu šedi pixelů objektů v popředí, vzhledem k jejich vzdálenosti k pixelu pozadí, tj. pixelu, který má hodnotu 0. V OpenCV je tato funkce implementována jako `distanceTransform`. Pro lepší vizualizaci je poté rozsah hodnot pixelů změněn na hodnoty od 0 do 1,0 pomocí OpenCV funkce `normalize`.

Následuje další zpracování pomocí prahování. Správným binárním prahováním jsou získány středy oblastí objektů, které budou sloužit jako značky(marker) pro Watershed transformaci. Po prahování je použita mírná dilatace, pro mírné zvětšení těchto značek.

Poté jsou nalezeny kontury všech značek v popředí. Tohoto je dosaženo pomocí OpenCV funkce `findContours`. Parametry funkce `findContours`:

- **image** - vstupní obraz
- **contours** - detekované kontury, každá kontura je uchována jako vektor bodů(Point), výstupem je vektor těchto vektorů
- **mode** - určuje, které kontury mají být nalezeny a jaká má být vytvořena hierarchie
- **method** - metoda aproximace kontury

Nalezené kontury jsou poté vykresleny do obrazu, který slouží jako mapa značek potřebné pro Watershed transformaci. Každá kontura je vykreslena jinou intenzitou šedi. Knihovna OpenCV obsahuje pro vykreslení funkci *drawContours*.

Po vykreslení kontur je již obraz připraven k samotné Watershed transformaci. Knihovna OpenCV obsahuje naimplementovanou Watershed transformaci označenou jako funkce *watershed*. Tato funkce byla využita v systému.

Parametry funkce *watershed*:

- **image** - vstupní 8-bitový 3-kanálový obraz
- **markers** - vstupně-výstupní 32-bitový jednobitový obraz(mapa) značek se stejnou velikostí jako vstupní barevný obraz

V posledním kroku je každé různé segmentované oblasti přiřazena různá barva tak, aby bylo možné tyto oblasti snadno odlišit. Takto vybarvený obraz je poté převeden na obraz šedotónový pomocí OpenCV funkce *cvtColor* podobně jako v prvním kroku. Na obraz je následně aplikována mírná eroze pro vytvoření zřetelnější hranice mezi oblastmi.

Na Obrázku 9 jsou ukázány jednotlivá předzpracování před aplikací Watershed transformace, jak jdou postupně za sebou v programu až po získání značek a nakonec obraz po aplikaci Watershed transformace.

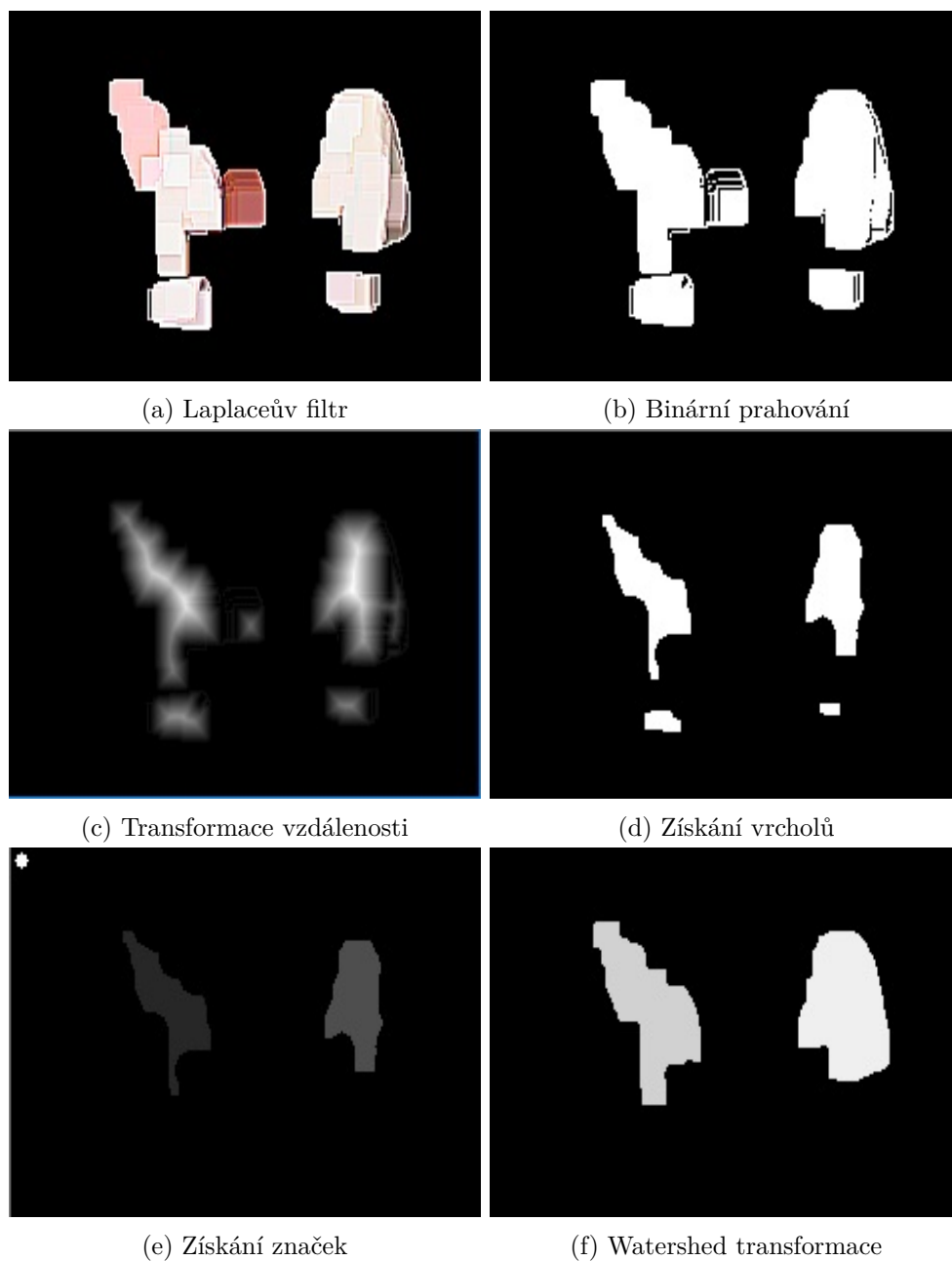
## 4.7 Vyhledání rukou

Po předchozím zpracování by v obrázku měly zůstat pouze oddělené odhalené části těla řidiče nebo spolujezdců a šum způsobený barvou interiéru a objektů v automobilu. V této sekci je popsáno, jakým způsobem jsou ruce řidiče v obrázku nalezeny.

Na předzpracovaný obraz je použita metoda k vyhledání kontur *findContours*, implementovaná v OpenCV. Je důležité, aby metoda vracela pouze vnější kontury, jelikož zpracování vnitřních kontur by mohlo způsobit chybnou detekci.

Nalezené kontury jsou poté filtrovány na základě jejich velikosti. Tímto jsou odstraněny malé kontury, které mohou být způsobeny šumem či okolními objekty. Pokud by nebyly odstraněny, je vysoce pravděpodobné, že by mohly být detekovány jako oblasti rukou v následujícím kroku.

Zbýlé kontury jsou poté porovnány a jsou z nich vybrány dvě největší kontury, které by ve zpracovaném obraze měly představovat oblasti rukou. Všechny ostatní kontury již nebudou používány a proto jsou zahozeny. Porovnání probíhá na základě obsahu kontur, vypočtené pomocí OpenCV funkce *contourArea*.



Obrázek 9: Jednotlivé kroky při aplikaci Watershed transformace

## 4.8 Určení pozice rukou

V této závěrečné části implementace je popsáno, jak je určena pozice rukou vzhledem k volant. Pro určení pozice je třeba stanovit vhodné parametry, které tuto polohu popíší. Výběr těchto parametrů je závislý na umístění kamery ve vozidle. Pro umístění kamery nad volantem byly vybrány tyto tři parametry:

1. Vzdálenost rukou od volantu
2. Úhel mezi rukama
3. Vzdálenost ruky od kamery

Pro určení některých z těchto parametrů je potřeba získat body, reprezentující střed volantu a těžiště oblastí rukou. Střed volantu je určen jako polovina velikosti obrazu, který byl získán vyznačením oblasti volantu z originálního obrazu před samotným zpracováním. Kvůli tomuto je velmi důležité, aby se volant nacházel co nejpřesněji uprostřed vyznačené oblasti. Těžiště oblastí rukou jsou vypočteny na základě momentů. Pro výpočet středu kontury slouží následující vztahy v Rovnici (29).

$$\begin{aligned} x &= \frac{m_{1,0}}{m_{0,0}} \\ y &= \frac{m_{0,1}}{m_{0,0}} \end{aligned} \tag{29}$$

kde  $m_{1,0}$  a  $m_{0,1}$  jsou vypočtené momenty vzhledem k osám, procházejícím těžištěm objektu a  $m_{0,0}$  reprezentuje plochu objektu. Pro následné určení úhlu a vzdálenosti je pro každou konturu vypočten vektor směřující od středu volantu ke středu kontury ruky. Vektor je vypočten vztahem popsaným v Rovnici (30).

$$\vec{v} = B - A \tag{30}$$

kde A je počáteční bod vektoru a B je koncový bod vektoru.

#### 4.8.1 Vzdálenost rukou od volantu

První z vlastností slouží ke zjištění, zda ruce jsou mimo oblast volantu či ne. Pokud ruka přesáhne vzdálenostní mez, nenachází se v oblasti volantu a tudíž nebude detekována. Vzdálenost je určena jako velikost vektoru, směřujícího od středu volantu do středu kontury ruky. Vztah pro výpočet je uveden v Rovnici (31).

$$|v| = \sqrt{v_1^2 + v_2^2} \tag{31}$$

kde  $v_1$  a  $v_2$  jsou souřadnice vektoru.

#### 4.8.2 Úhel mezi rukama

Druhá vlastnost určuje vzájemnou polohu rukou. Dle úhlu lze zjistit, zda se ruce nachází blízko sebe nebo se nachází na opačné straně volantu. Jestliže je volant držen dle předpisů, tak úhel mezi vektory vedoucí od středu volantu ke středům rukou by měl být spíše tupý. Pokud jsou

ruce u sebe, je pravděpodobnější, že úhel bude velmi ostrý. Určení úhlu je samozřejmě možné pouze v případě, že byly detekovány právě dvě ruce. Vztah pro výpočet úhlu mezi vektory je odvozen ze vztahu pro výpočet skalárního součinu vektorů v Rovnici (32).

$$\alpha = \arccos \left( \frac{\vec{u} \cdot \vec{v}}{|\vec{u}| \cdot |\vec{v}|} \right) \quad (32)$$

kde  $\vec{u}$ ,  $\vec{v}$  jsou vektory, svírající úhel  $\alpha$  a  $|\vec{u}|$ ,  $|\vec{v}|$  jsou velikosti těchto vektorů

### 4.8.3 Vzdálenost ruky od kamery

Poslední vlastností je vzdálenost ruky od kamery určená hodnotou hloubky v oblasti ruky v hloubkovém obraze. Tento parametr je ze všech tří nejdůležitější. Pomocí této vlastnosti je určeno, v jaké oblasti volantu se ruce nacházejí. Nejdříve je třeba určit horní a dolní hranici hloubky, nacházející se nad a pod volantem. Horní hranice je poté odečtena od stanovené dolní hranice. Tímto je získán rozsah hloubky pro celou oblast volantu. Tento rozsah je poté rozdělen na několik sektorů. Dle vypočtené průměrné hloubky nalezené oblasti ruky, získané ze vstupní 16-bitové hloubkové mapy, je poté určeno, do kterého sektoru oblasti volantu ruka patří. Pokud se ruka řidiče dostane do spodní části volantu, pak systém zaznamená špatnou pozici rukou.

## 5 Experimenty a výsledky

Systém byl vyvíjen a testován na operačním systému Windows 10 64-bit, procesor Intel Core i7-6700HQ, 8GB RAM. Testování bylo provedeno na čtyřech videích s různými osobami a v různých prostředích. Na posledním videu byla testována funkčnost metody pro osoby s odhalenými pažemi. Z každého videa bylo otestováno 150 snímků. Vstupní videa byla pořízena v kancelářském prostředí s herním volantem pomocí kamery Kinect V2, která byla umístěna do vhodné pozice a výšky tak, aby simulovala možné umístění nad čelním sklem v reálném automobilu. Video pak obsahovala pohled na interiér vozidla a především oblast volantu a řidiče.

Jelikož použitá metoda je založena na segmentaci oblastí s barvou kůže, tak je samozřejmé, že použitá metoda nemůže být aplikována, pokud řidič nosí při řízení rukavice, jelikož zakryjí celou potenciálně detekovatelnou oblast. Model je zároveň navržen tak, že nezahrnuje řidiče, kteří mají velmi tmavou pleť.

V tabulce 1 jsou popsány statické hodnoty, které bylo nutno určit pro každé různé testovací prostředí zvlášť.

Název parametru	Video 1	Video 2	Video 3	Video 4
Hodnota prahu pro odstranění pozadí	12000	13000	14000	14000
Hranice hloubky pod volantem	12000	11500	12000	12000
Hranice hloubky nad volantem	9000	9800	10000	10000

Tabulka 1: Nastavené statické hodnoty prahu a hloubky

Výpočet přesnosti detekce je uveden v Rovnici (33).

$$Přesnost[\%] = \frac{\text{počet správných detekcí}}{\text{celkový počet snímků}} \cdot 100 \quad (33)$$

O správnou detekci se jedná v případě, že ruka byla detekována v oblasti volantu a bylo správně určeno, zda se nachází ve spodní či v horní části volantu nebo zda jsou ruce příliš blízko sebe. Pokud ruce nejsou v oblasti volantu a nejsou detekovány, tak se jedná taktéž o správnou detekci. V tabulce 2 jsou uvedeny nejčastější problémy, vyvolávající chybnou detekci, přesnosti detekcí pro každou testovanou osobu a celková průměrná přesnost detekce.

	Video 1	Video 2	Video 3	Video 4
Celkový počet snímků	150	150	150	150
Počet správných detekcí	125	126	127	116
Počet nenalezení obou rukou	5	19	17	20
Počet vyhodnocení objektu interiéru jako ruky	16	3	0	0
Počet chybných stanovení polohy rukou	4	2	6	14
Přesnost[%]	83,33	84	84,67	77,33
Průměrná přesnost[%]	82,33			

Tabulka 2: Přesnost detekce

Na Obrázku 10 jsou zobrazeny úspěšné detekce rukou při jejich správné poloze na volantu. Správnou polohu rukou značí zelené obdélníky.



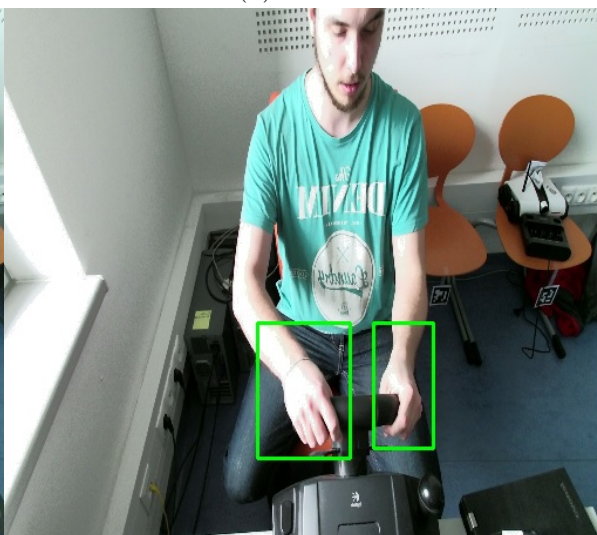
(a) Video 1



(b) Video 2



(c) Video 3

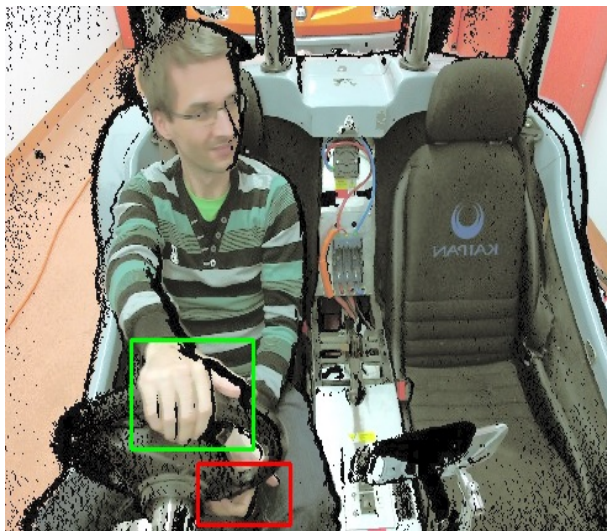


(d) Video 4

Obrázek 10: Ukázka správné detekce rukou při jejich správné pozici



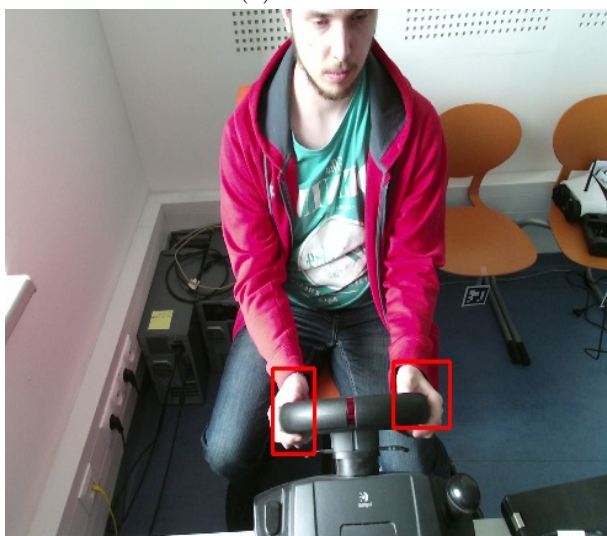
Na Obrázku 11 jsou zobrazeny úspěšné detekce rukou při jejich špatné pozici na volantu. Pokud jsou ruce při držení volantu příliš nízko, tak jsou označeny červeným obdélníkem. Pokud se ruce ocitnou na volantu příliš blízko sebe, tak vypočtený úhel mezi nimi je velmi ostrý a jsou poté označeny modrým obdélníkem.



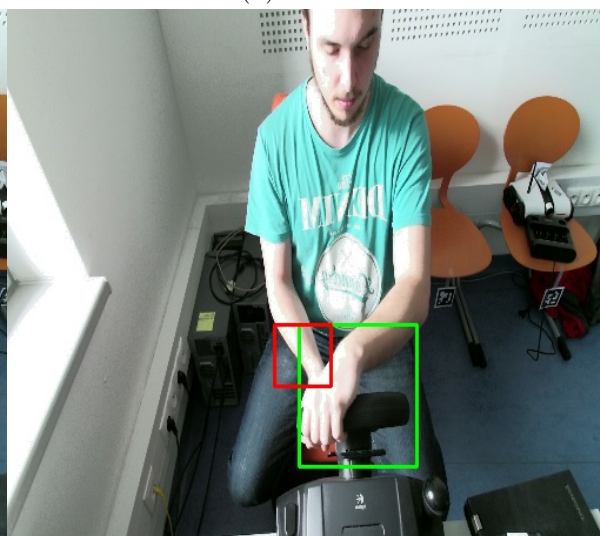
(a) Video 1



(b) Video 2



(c) Video 3



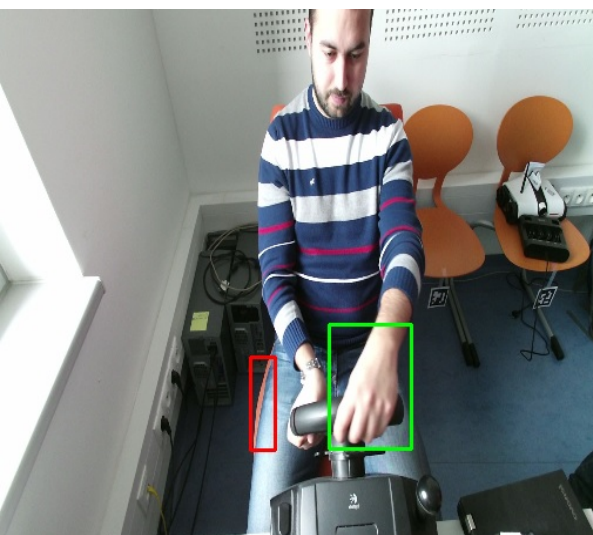
(d) Video 4

Obrázek 11: Ukázka správné detekce rukou při jejich špatné pozici

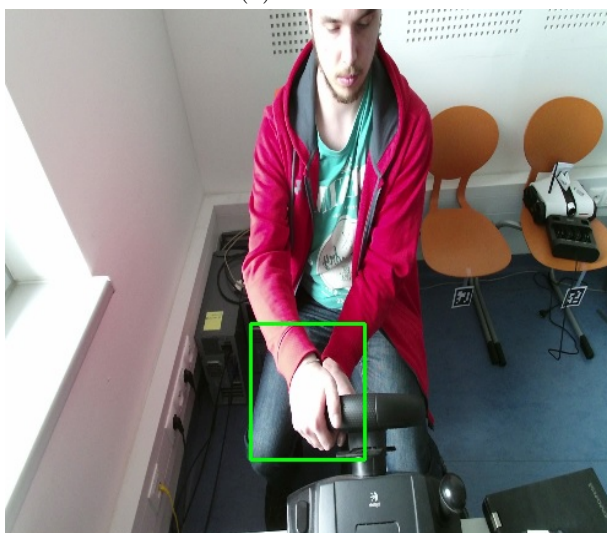
Na Obrázku 12 jsou zobrazeny ukázky neúspěšné detekce rukou nebo jejich špatné určení polohy. Na prvním obrázku je ruka rozdělena volantem na dvě poloviny. Druhá ruka mimo dosah detekce. Vznikly dvě malé oddělené oblasti, které byly vyhodnoceny jako dvě ruce. Na druhém obrázku byla židle vyhodnocena jako ruka, jelikož její barva je podobná barvě kůže. Na třetím obrázku jsou ruce těsně nad sebou a proto splynuly v jednu oblast. Na čtvrtém obrázku ostré bílé světlo snížilo přesnost segmentace kůže a došlo k chybnému vyhodnocení polohy, jelikož do výpočtu byla zahrnuta ruka ve spodní části volantu.



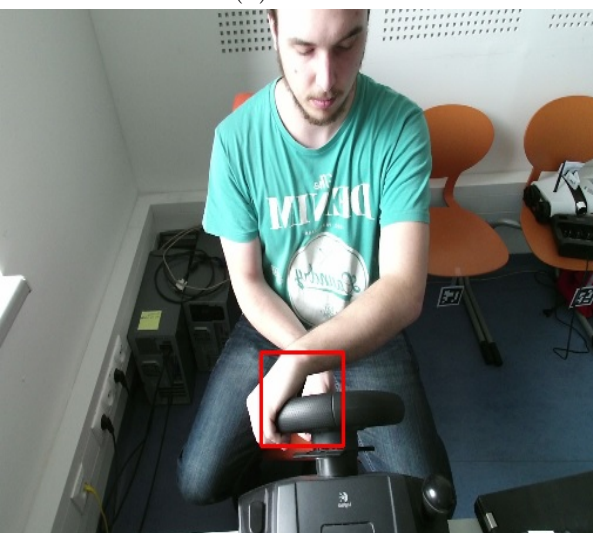
(a) Video 1



(b) Video 2



(c) Video 3



(d) Video 4

Obrázek 12: Ukázka chybných detekcí nebo špatné určení polohy rukou

## 6 Závěr

Úkolem této bakalářské práce bylo jednak najít informace o vhodných metodách, které slouží pro detekci a určení pozice rukou na volantu, a jednak vytvořit systém, který by jednu z těchto metod využil. Na základě získaných poznatků byla vybrána metoda, založená na segmentaci kůže. Následně byl vytvořen algoritmus, který je schopen detekovat a určit polohu rukou. Algoritmus obsahuje důležité části zpracování videa od samotného načtení videa, po jeho předzpracování, segmentaci a následnou detekci a určení polohy rukou.

Veškeré teoretické poznatky potřebné k vlastnímu návrhu řešení jsou uvedeny v teoretické části. Tato část se věnuje zejména barevným modelům, ze kterých jsou následně vyhrazeny oblasti, které definují barvu kůže, a pomocí těchto oblastí jsou pro každý model určeny omezující pravidla pro segmentaci kůže z obrazu. Dále jsou v této části obsaženy informace o použitých technikách, kterými jsou prahování, Watershed transformace a vyhledání kontur.

V praktické části je popsán výsledný navržený algoritmus, jeho implementace a funkce s ním spojené, včetně nastavených omezujících pravidel pro barvu kůže. Dále je v ní obsaženo testování a výsledky implementovaného algoritmu. Jsou zde ukázány výsledné obrazy s provedenou detekcí a je zde poukázáno na chyby, které mohou nastat. Dle testů je metoda nepříliš efektivní při silném bílém světle a pokud by se v oblasti volantu nacházely objekty stejné velikosti a barvy jako mají ruce řidiče. Dalším problémem je, že úspěch této metody je značně závislý na nastavených parametrech. Důležitá je samozřejmě úspěšnost této metody, která činila 82,33 procent.

Aplikace by dále mohla být rozšířena o uživatelské rozhraní, které by umožnilo nastavit vhodné parametry pro detekci rukou v různých prostředích a podmínkách. Dále by mohla být rozšířena o algoritmus, který by byl schopen automaticky vyhledat oblast volantu, a tím odstranit potřebu ohrazení oblasti samotným uživatelem.

Tato práce nás uvádí do problematiky detekce rukou na volantu ve vozidle a navržený algoritmus tvoří základ pro složitější řídicí systémy v různých průmyslových oblastech u zařízení, vyžadujících detekci polohy rukou obsluhujícího operátora. Jedním z možných použití je v automobilech se samořídícím systémem, kde by metoda mohla být využita pro přepínání mezi manuálním řízením řidiče a samořízením automobilu.

## Literatura

- [1] “A primer to colors in digital design.” <https://uxdesign.cc/a-primer-to-colors-in-digital-design-7d16bb33399e>. [Online; 19. dubna 2018].
- [2] “Computational color.” <http://printingcode.runemadsen.com/lecture-color/>. [Online; 19. dubna 2018].
- [3] S. Kolkur, D. Kalbande, P. Shimpi, C. Bapat, and J. Jatakia, “Human skin detection using rgb, hsv and ycbcr color models,” *arXiv preprint arXiv:1708.02694*, 2017.
- [4] T. Zhou, P. J. Pillai, and V. G. Yalla, “Hierarchical context-aware hand detection algorithm for naturalistic driving,” in *Intelligent Transportation Systems (ITSC), 2016 IEEE 19th International Conference on*, pp. 1291–1297, IEEE, 2016. ISBN: 978-1-5090-1889-5.
- [5] O. Lapáček, “Využití knihovny opencv pro zpracování obrazu v mobilních robotických systémech,” 2009.
- [6] “Understanding the rgb color model.” <https://www.lifewire.com/color-models-rgb-1697461>. [Online; 23. března 2018].
- [7] “What is the hsv color model?.” <https://www.lifewire.com/what-is-hsv-in-design-1078068>. [Online; 23. března 2018].
- [8] “Color conversion.” <http://www.equasys.de/colorconversion.html>. [Online; 23. března 2018].
- [9] “About depth map.” [http://www.i-art3d.com/Eng/About\\_Depth.htm](http://www.i-art3d.com/Eng/About_Depth.htm). [Online; 6. dubna 2018].
- [10] E. Sojka, “Digitální zpracování a analýza obrazu,” 2000.
- [11] “Oficiální dokumentace knihovny opencv.” <https://docs.opencv.org>. [Online; 14. března 2018].
- [12] A. Dhawan and V. Honrao, “Implementation of hand detection based techniques for human computer interaction,” *arXiv preprint arXiv:1312.7560*, 2013.
- [13] D. G. Bailey, “An efficient euclidean distance transform,” in *International workshop on combinatorial image analysis*, pp. 394–408, Springer, 2004. ISBN: 978-3-540-30503-3.
- [14] J. B. Roerdink and A. Meijster, “The watershed transform: Definitions, algorithms and parallelization strategies,” *Fundamenta informaticae*, vol. 41, no. 1, 2, pp. 187–228, 2000.
- [15] N. Amoda and R. K. Kulkarni, “Image segmentation and detection using watershed transform and region based image retrieval,” *Int. J. Emerg. Trends & Techno. Comp. Sci*, vol. 2, pp. 89–94, 2013.

- [16] N. A. bin Abdul Rahman, K. C. Wei, and J. See, “Rgb-h-cbcr skin colour model for human face detection,” *Faculty of Information Technology, Multimedia University*, vol. 4, 2007.
- [17] J. Kovac, P. Peer, and F. Solina, *Human skin color clustering for face detection*, vol. 2. IEEE, 2003. ISBN: 0-7803-7763-X.

## Přílohy

DVD, které obsahuje:

- Projekt se zdrojovými kódy
- Elektronická forma textu bakalářské práce
- Testovací videa